Problem 1: kNN Classifier (30 points)

1) Use the leave one out cross validation on the training data to select the best k among $\{1, 2, \ldots, 10\}$. Report the averaged leave-one-out error (averaged over all training data points) for each $k \in \{1, 2, \ldots, 10\}$.

2) Based on 1), use the best k to predict the class labels for test instances. You should also report the predicted labels for the testSet.

See hw3_code/p1.py → produces the following output:

```
(base) samuelfafel@Fafels-MBP hw3_code % python3 p1.py
Loading data...
        trainX.shape (243, 13)
        trainY.shape (243,)
        testX.shape (27, 13)
Completing 'Leave-One-Out' Cross Validation...
        k: error                (percent)
        1: 0.22633744855967075  (22.63%)
        2: 0.22633744855967075  (22.63%)
        3: 0.1728395061728395   (17.28%)
        4: 0.18518518518518523  (18.52%)
        5: 0.18106995884773658  (18.11%)
        6: 0.16460905349794241  (16.46%)
        7: 0.16872427983539096  (16.87%)
        8: 0.18106995884773658  (18.11%)
        9: 0.18106995884773658  (18.11%)
        10: 0.17695473251028804 (17.70%)
        Best k = 6
Predicting Test Labels...
        [-1.0, -1.0, 1.0, -1.0, 1.0, -1.0, -1.0, 1.0, -1.0, 1.0, 1.0, 1.0, -1.0, -1.0, -1.0, 1.0, 1.0, -1.0, -1.0, -1.0, 1.0, 1.0, 1.0, 1.0, -1.0, -1.0, -1.0]
(base) samuelfafel@Fafels-MBP hw3_code %
```

Problem 2: PCA (50 points)

1) Train a kNN based on the original features. You should conduct cross-validation (of your choice) to select the best k. Describe the cross-validation approach, e.g., how many folds, how did you split the data. If you use a library for cross-validation, describe how the library does it. Report the best value of k under your cross-validation approach, and then report the testing accuracy corresponding to the best k.

2) Conduct PCA on the data and then learn a kNN model using the dimensionality reduced data. You should conduct cross-validation (of your choice) to select the best k and best d, where d is number of dimensions after PCA. Describe the cross-validation approach and report the best value of k and best value of d. Report the testing accuracy corresponding to the best k and best d.

3) Visualize the data distribution in the PCA projected space with d=1, d=2, and d=3, respectively. Include the visualization as figures in your pdf. Are there any visible clusters or patterns in the PCA plot? How does the PCA visualization change with different numbers of components?

1. First kNN and CV:
    a. Cross-validation approach: I used a 5-fold cross-validation approach, testing values for *k* in the range 1 to 20.
    b. Library for cross-validation: I used the "sklearn" python library for cross-validation. This library first splits the data into folds, then loops over those folds. Each iteration one fold is used as the validation set while the rest make up the training set. The model is trained on the training set and then evaluated on the test set. The model is evaluated by its accuracy (proportion of correct predictions). After all iterations are complete, the function returns an array of scores (one for each fold).
    c. Best value of *k*: 4
    d. Testing Accuracy for $k = 96.70\%$:
2. PCA, second kNN and CV:
    a. Cross-validation approach: 5-fold cross-validation, for *k* in range 1-20, *d* in range
    b. Best value of *k*: 5
    c. Best value of *d*: 19
    d. Testing Accuracy for $k = 5$, $d = 19$: 97.18%

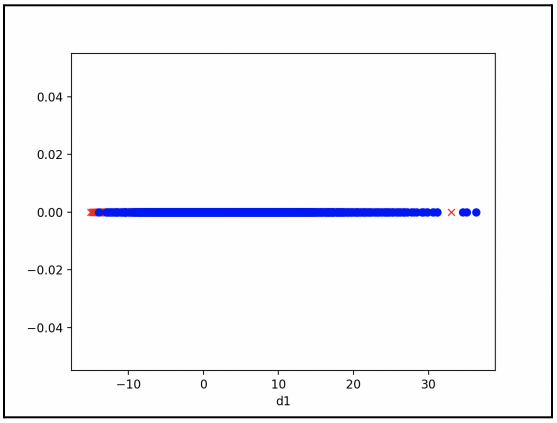
3.

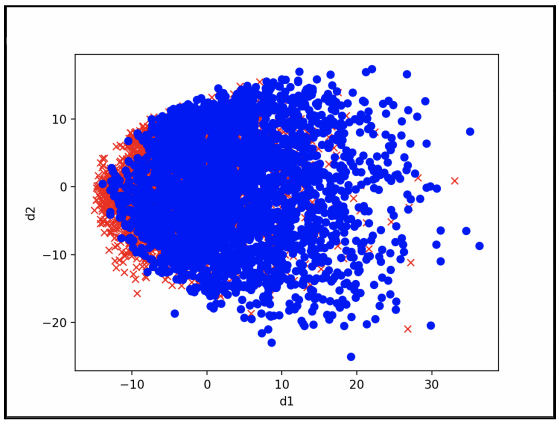
a. Graphs:

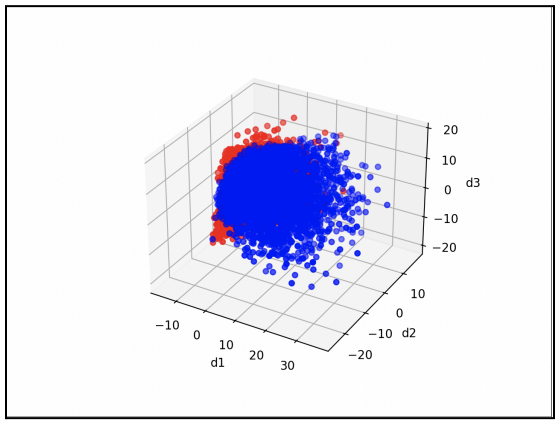
i. k = 5, d = 1: "*IndexError: index -2 is out of bounds for axis 1 with size 1*"

ii. k = 5, d = 2: "*IndexError: index -3 is out of bounds for axis 1 with size 2*"

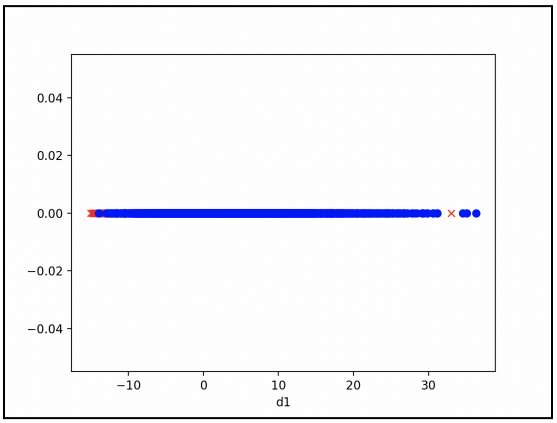
iii. k = 5, d = 3:

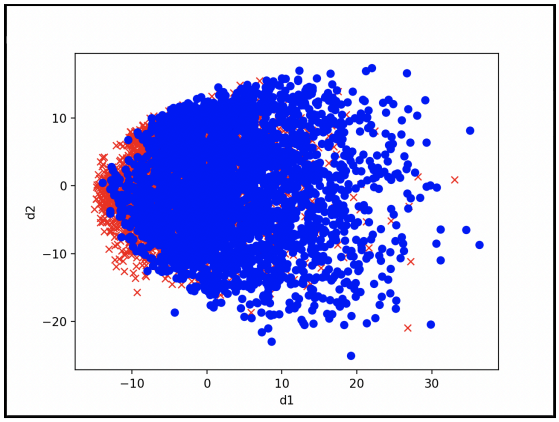


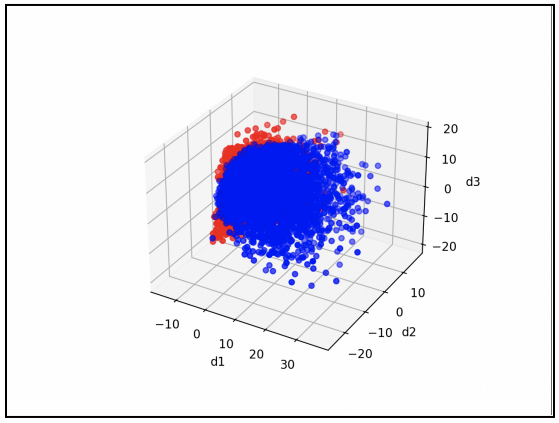




iv. k = 5, d = 19:







b. Visible clusters/Patterns:
   i. There are two clusters visible within the data: red and blue. Depending on the way the graph is viewed, both appear approximately spherical.
c. How does PCA visualization change with different numbers of components?
   i. There is little to no change in the visualization as *d* switches from 3 to 19. For *d* = 1 and *d* = 2, the provided code was unable to visualize the data. (Provided index errors)

Problem 3: (20 points)

Given a set of observations $\{x_1, x_2, ..., x_n\}$, where $x_i \in \{1, 2, ..., K\}$. Assume that each $x_i, i = 1, ..., n$ follows an identical and independent distribution specified by $Pr(x_i = k) = p_k$, where $\sum_{k=1}^{K} p_k = 1$. Please derive the maximum likelihood estimation of the parameter $p = (p_1, p_2, ..., p_K)$. Hint: You can use a one-vs-rest strategy. For example, when you consider p_1, x_i can be viewed as either 1 or not 1.

Likelihood = $\prod_{i=1}^{n} \Pr\left(x_i = k\right)^{I\left(x_i = k\right)}$ where I is the indicator function (1 if $x_i = k$ else 0)

log likelihood = $\sum_{i=1}^{n} \sum_{k=1}^{K} I\left(x_i = k\right) \log\left(p_k\right)$

$$L(p, \lambda) = \sum_{i=1}^{n} \sum_{k=1}^{K} I\left(x_i = k\right) \log\left(p_k\right) + \lambda\left(1 - \sum_{k=1}^{K} p_k\right)$$

$$\frac{\partial L}{\partial p_k} = \frac{1}{p_k} \sum_{i=1}^{n} I\left(x_i = k\right) - \lambda = 0 \Rightarrow \sum_{i=1}^{n} I\left(x_i = k\right) = \lambda p_k$$

Using the constraint $\sum_{k=1}^{K} p_k = 1$, we find that $n = \lambda$

Thus, $p_k = \frac{1}{n} \sum_{i=1}^{n} I\left(x_i = k\right)$

This says that the maximum likelihood estimate for each p_k is the proportion of the observations that equal k.